AD-A049 344 GEORGIA INST OF TECH ATLANTA SCHOOL OF INFORMATION A--ETC F/G 12/1
AN EMBEDDING RESULT FOR LABELLED PROGRAMS.(U)
JUN 77 R A DEMILLO, S R KOSARAJU DAAG29-76-G-0338
UNCLASSIFIED GIT-ICS-77/09 ARO-14690.2-M NL

1 OF 1
AD
A049344

END
DATE
FILMED
3-78
DDC

(14) GIT-ICS-77/09

(9) TECHNICAL REPORT

(6)

AN EMBEDDING RESULT FOR LABELLED PROGRAMS *

(11) Jun 77

(10) Richard A. DeMillo*
School of Information and Computer Science
Georgia Institute of Technology
Atlanta, GA 30332

(12) 15p.

S. Rao Kosaraju**
Department of Electrical Engineering
Johns Hopkins University
Baltimore, MD 21218

(18) ARO

(19) 14690.2-M

(15)

* ~~Supported in part~~ by the U.S. Army Research Office, Grant Nos. DAAG29-76-G-0338 ~~and DAAG29-74-G-0179~~.

** ~~Supported in part~~ by the National Science Foundation, Grant No. NSF-DCR75-09904

410 044

[illegible]

INTRODUCTION

There are two natural methods of limiting the use of labels in structured programs: bounding the number of labels that can be referenced by a single statement and bounding the total number of labels which can appear in a program. It is implicit in an argument of [1] in the former case and in the unbounded analog of both cases that a genuine limitation *is* imposed and power increases with the number of labels. We show here that, in the latter case, programs with differing bounded numbers of labels are provably equivalent in the precise sense of [1,2]. From [3] it is known that suitable restrictions on the notion of equivalence result in provable differences among these constructs; these restrictions, however, rely on the details of program organization. Hereafter, we deal only with combinatorial arguments. Further motivation for the combinatorial properties in the sequel may be found in [1].

PRELIMINARIES

A directed graph G is composed of a set of *vertices*, $V(G)$, and *arcs* $E(G) \subset V(G) \times V(G)$. The arcs (x,y) and (y,x) together form an *edge* of G. Arcs and edges are represented by directed and undirected arrows, respectively. A *path* from x to y is a sequence of arcs

$$(x, x_1), (x_1, x_2), \ldots , (x_{n-2}, x_{n-1}), (x_{n-1}, y),$$

and such a path is said to be of *length n*. We define the distance metric $d_G(x,y)$ to be the minimum of the lengths of all paths from x to y.

A *binary tree* with root x is a directed graph that is either a single vertex x or contains a vertex x connected by *edges* to root(s) y_i of subtree(s) G , $i \leq 2$. Note that d_G is symmetric on binary trees. Let G be a binary tree with root x_0 and consider the path $(x_0, x_1), \ldots, (x_{n-1}, x_n)$, where $x_i \neq x_j$ for $i \neq j$. We define the following relations on G:

(1) x_j is a *descendent* of x_i $(0 \leq i < j \leq n)$

(2) x_i is an *ancestor* of x_j $(0 \leq i < j \leq n)$

(3) x_i is the *father* of x_{i+1} $(0 \leq i < n)$, and we write $x_i = f_G(x_{i+1})$

(4) x_{i+1} is a *son* of x_i $(0 \leq i < n)$

(5) x_n is *leaf* of G if $f_G(y) \neq x_n$ for all $y \in V(G)$

In a binary tree G, the subtree with root x is denoted by G_x.

An *ancestor tree* is a directed graph G whose arcs may be partitioned into two maximal subsets E_1, E_2 such that $(V(G), E_1)$ is a binary tree and if $(x,y) \in E_2$, then y is an ancestor of x in the binary tree $(V(G), E_1)$. Thus, in an ancestor tree a vertex may be connected by an arc to any of its ancestors. We use the special notation $x \underset{a}{\rightarrow} y$, if $(x,y) \in E_2$. The following terminology is suggested by [1]: if $y \underset{a}{\rightarrow} x$ then x is a *label* and y is an *exit*.
We then say that an ancestor tree, G, is a *k-label program* if it contains at most k-labels; we also say that G is a *k-exit program* if it contains at most k-exits.

SPACE-TIME BOUNDED SIMULATIONS

The following definition is from [1]; it introduces a fundamental mechanism for comparing programs. Let G, G* be directed graphs. We say that G* *simulates* G with *space* dilation $S > 0$ and *time* dilation $T > 0$, written $G \leq_{S,T} G^*$ if there is a map (called an *embedding* of G in G*).

$$\Phi: V(G^*) \rightarrow V(G) \cup \{\Lambda\},\ \Lambda \notin V(G) \cup V(G^*)$$

such that:

$$(1)\quad \forall u \in V(G)$$

$$0 < |\Phi^{-1}(u)| \leq S ,$$

and

$$(2)\quad \forall v^* \in V(G^*) \text{ such that } \Phi(v^*) \neq \Lambda$$

$$\forall w \in V(G) \text{ such that}$$

$$d_G(\Phi(v^*), w) < \infty$$

$$\exists w^* \in V(G^*) \text{ such that } \Phi(w^*) = w, \text{ and}$$

$$d_{G^*}(v^*, w^*) \leq T \cdot d_G(\Phi(v^*), w) .$$

If Φ is an embedding and $\Phi(u^*) = \Lambda$, then u* is said to be a *bookkeeping* vertex; on the other hand, if $\Phi(u^*) = u \neq \Lambda$, then u* is said to be a *copy* of u. Clearly, in a simulation of G with space dilation S, no vertex of G can have more than S copies in the preimage of the embedding. In the sequel, we will avoid some notational unpleasantness by agreeing that $\lambda_1, \lambda_2, \ldots$ always denote bookkeeping vertices and that $u_1^*, u_2^*, \ldots u_k^*$, $k \leq S$, always denote copies u.

It is known that for every S, T > 0, there is an ancestor tree which cannot be simulated with space and time dilation S and T by any 1-exit program and for every S, T there is a 1-exit program which cannot be simulated with space and time dilation S and T by any 1-label program. This is very suggestive of a hierarchy in the number of labels for the $\leq_{S,T}$ relation, among ancestor trees.

We can now show that such hierarchies collapse. That is, we show that for every $k > 1$, there is a $T \geq 0$ such that every k-label program can be simulated by some 1-label program with space dilation $S = 1$ and time dilation $T = T(k)$. We begin by considering a general embedding procedure which dilates space by $S(k) > 1$, since this result is technically easier.

AN OBSERVATION

Let G be a binary tree with root x and consider a vertex y which is not the father of two vertices. G can be modified by viewing y as the root and inverting the father-son relationship along the path from x to y. Obviously, the resulting graph is still a binary tree; we denote this tree by G^y.

MAIN SIMULATION RESULT

Let H be an ancestor tree and choose a vertex x of H such that among the descendents of x there is exactly one label y. Let H' be obtained from H by replacing the subtree H_x by the graph shown in Figure 1.

In this graph, K is H_x with its vertices subscripted by "1", L is $(H_x - H_y)^{f_H(y)}$ with its vertices subscripted by "2", and M is H_y with its vertices subscripted by "2". Thus α is a "second copy" of $f_H(y)$. In addition, each arc $u \underset{a}{\rightarrow} y$ or $u \underset{a}{\rightarrow} x$ is replaced by $u_1 \underset{a}{\rightarrow} \lambda_2$, $u_2 \underset{a}{\rightarrow} \lambda_2$, while every $u \underset{a}{\rightarrow} v$ with $v \neq x,y$ is replaced by $u_1 \underset{a}{\rightarrow} v$ and $u_2 \underset{a}{\rightarrow} v$. Then we have $H \leq_{2,3} H'$, which may be proved easily by a case analysis of the possible arcs in H and their copies in H'. Note further that if x is not a label, then H' has the same number of labels as H, while if x is a label, the total number of labels is decreased by one.

We now prove that any k-label program $(k \geq 2)$ can be simulated by a (k-1)-label program. To this end, let H be a k-label program, $k \geq 2$. Two cases arise.

<u>Case I.</u> Some vertex x contains exactly one label in each of its subtrees. If the root of either subtree is a label, no transformation is required for that subtree. In all other cases, replace each subtree as above to yield a k-label program H', where $H \leq_{2,3} H'$ and in H' the sons of x are both labels. Let the sons of x by $y,z \in V(H')$. Now, clearly each arc $u \underset{a}{\rightarrow} y$ or $u \underset{a}{\rightarrow} z$ can be replaced by an arc $u \underset{a}{\rightarrow} x$ at the expense of dilating path lengths by one arc. Hence, this transformation has the effect of replacing the pair of labels y,z by a single label x. If H" is the result of such a transformation, then since $d_{H''}(x,y) = d_{H''}(x,z) = 1$, we have $H \leq_{2,3} H''$.

<u>Case II.</u> Some label vertex x has exactly one label y as a descendent. The transformation given above when applied to the subtree rooted at x yields a k-1 label program H' such that $H \leq_{2,3} H'$.

Thus, every k-label program is simulated with S = 2, T = 4 by a k-1 label program. We have immediately that every k-label program H is simulated by some 1-label program G with S, T independent of $|V(H)|$; more specifically

$S = 2^{k-1}$, $T = 4^{k-1}$. It is easily seen that for every S, T there is a 1-label program which cannot be simulated by any 0-label program (i.e., by a binary tree).

AN IMPROVEMENT

The vertex duplication in the construction above is somewhat artificial; it is used only to keep track of "end points" of circuits, and we might try to use some inherent symmetry in the problem to avoid such duplication. In fact, such duplication need never be introduced. That is, we can prove that for every k-label program H, ($k \geq 2$) there is a (k-1)-label program H' such that $H \leq_{1,4} H'$.

Let B_n denote the regular graph on n vertices with degree 2, shown in Figure 2. If $V(B_n) = \{a_1, \ldots, a_n\}$, then $B_n \leq_{1,3} G_0$; and $B_n \leq_{1,4} G_1$, when G_0 and G_1 are as shown in Figures 3(a) and 3(b), respectively.

The first simulation is apparently the better of the two, but in fact the simulation $H \leq_{1,4} G_1$ is the one which is to be preferred for the simulation to be described.

Using this transformation, given a tree H, as shown in Figure 4(a), we embed $H \leq_{1,4} H^*$ where H^* is as in Figure 4(b). As in the case $B_n \leq_{1,4} G$, we now have a_1 and a_n relatively close to each other. For obvious reasons, we call this transformation a *folding* of H.

Now suppose that H is a subtree of a k-label program $H_0 (k \geq 2)$ that a_1 and a_n are both labels, $V(H_1)$ has no other label, and none of $a_2, \ldots, a_{n-1}$ is a label. We then form H' by folding H and replacing each $u \underset{a}{\rightarrow} a_n$ by $u \underset{a}{\rightarrow} a_1$. If no such subtree H of H_0 exists, then no label is related to any other as either a descendent or an ancestor. But since each $\{x,y\} \subseteq V(H_0)$ share a common ancestor (viz. the root of H_0) choose any two labels x,y and let z be their deepest common ancestor. This identifies subtrees of the form H with $a_1 = z$ and $a_n \in \{x,y\}$. Fold each of these subtrees and replace each arc $u \xrightarrow[a]{} x$ or $u \xrightarrow[a]{} y$ by $u \xrightarrow[a]{} z$. Then, if the resulting ancestor tree is H', we have $H_0 \leq_{1,4} H'$.

Note that the passage from a k-label program to a 1-label program still requires $T = 4^{k-1}$. It is not known if this is (asymptotically) the best possible.

REFERENCES

1. R. Lipton, S. Eisenstat, R. DeMillo, "Space and Time Hierarchies for Classes of Control Structures and Data Structures", *Journal of the ACM,* Vol. 23, No. 4, Oct 1976, pp. 720-732.

2. R. DeMillo, S. Eisenstat, R. Lipton, "Space-Time Tradeoffs in Structured Programming: An Improved Combinatorial Embedding Theorem" (to appear); parts of this paper appear as "Space-Time Tradeoffs in Structured Programming", *Proceedings of the 1976 Johns Hopkins Conference in Information Sciences and Systems*, pp. 431-434.

3. S. R. Kosaraju, "Analysis of Structure Programs", *Journal of Computer and System Sciences*, Vol. 9, No. 3, Dec 1974, pp. 232-255.

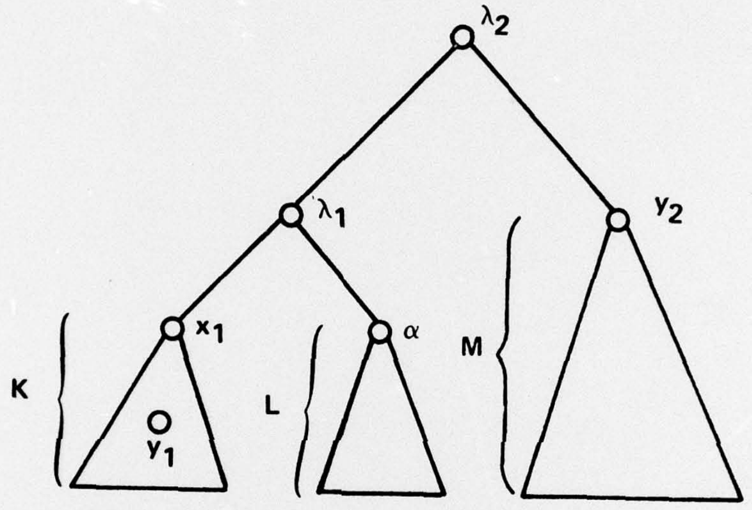


Figure 1. Modification of H

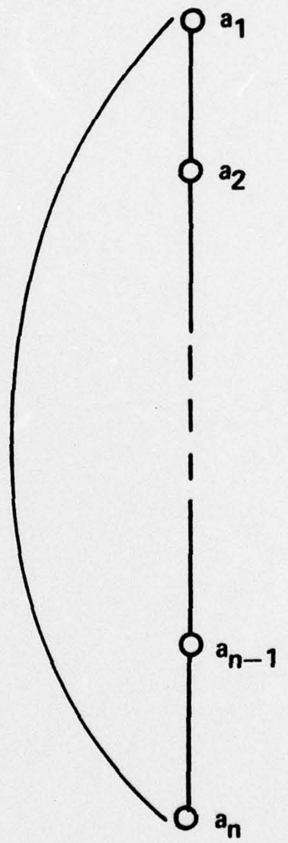


Figure 2. The Graph B_n

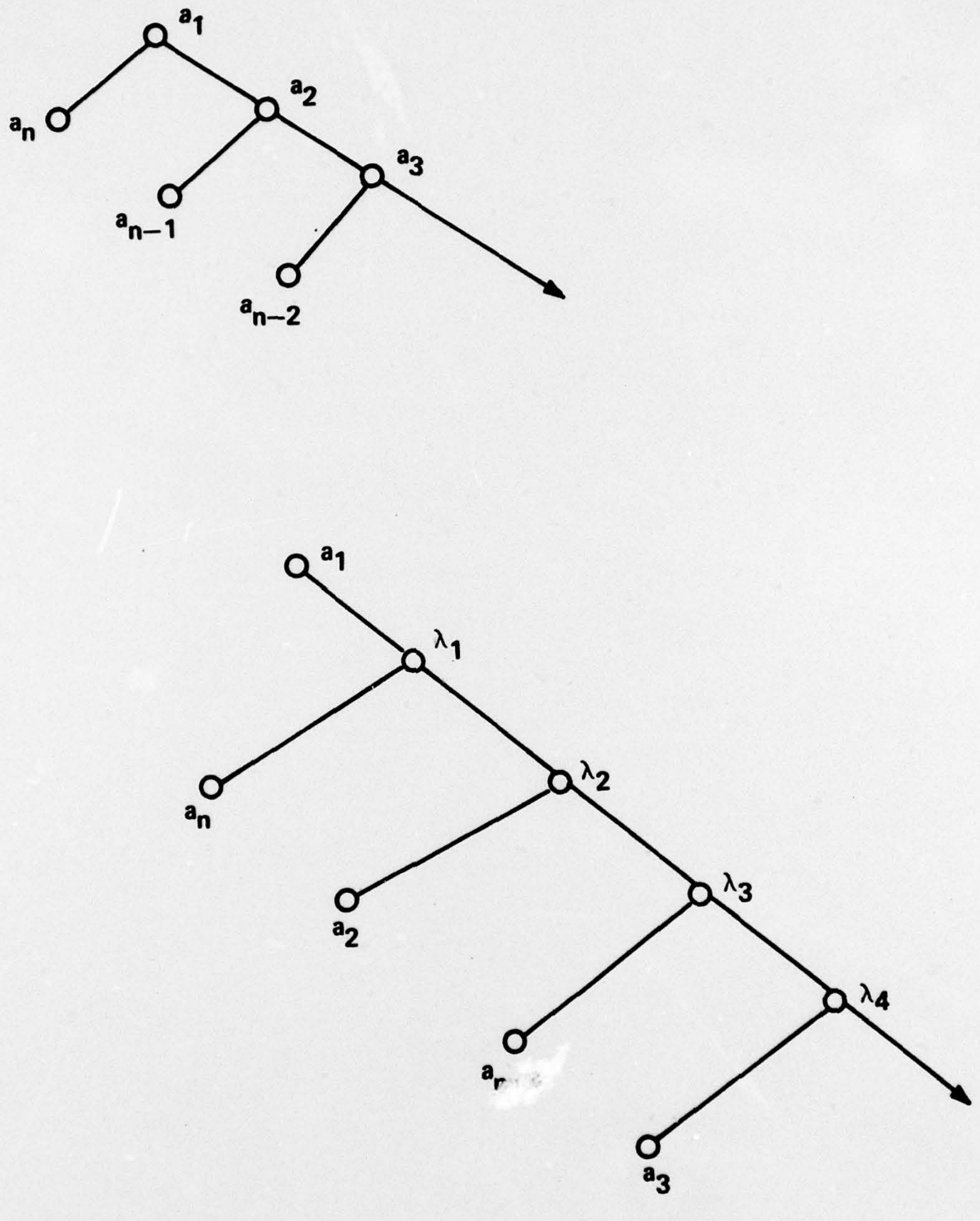
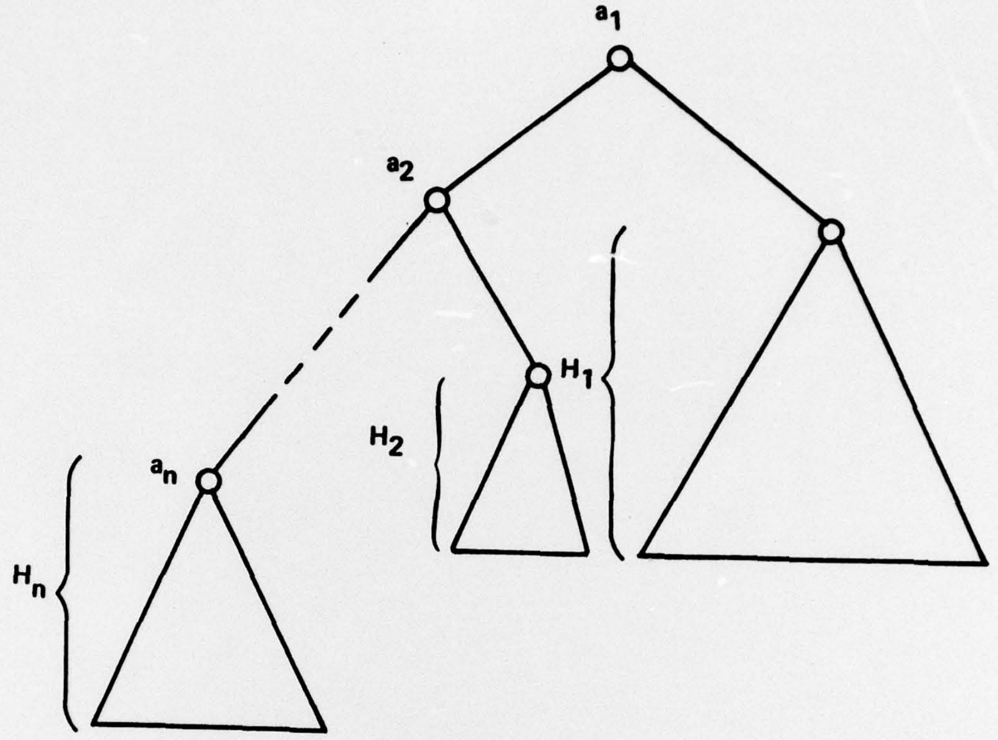


Figure 3. The Graphs G_0 (upper) and G_1 (lower)



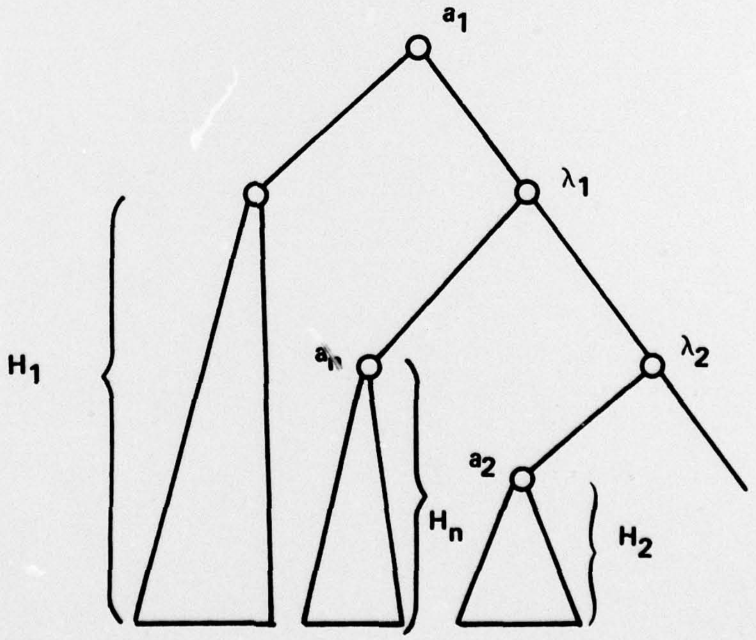


Figure 4. The trees H (upper) and H* (lower)

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered) UNCLASSIFIED

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER GITICS77/09 — 2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) An Embedding Result For Labelled Programs	5. TYPE OF REPORT & PERIOD COVERED Technical Report
	6. PERFORMING ORG. REPORT NUMBER GITICS77/09
7. AUTHOR(s) Richard A. DeMillo, S. Rao Kosaraju	8. CONTRACT OR GRANT NUMBER(s) DAAG29-76-G-0338
9. PERFORMING ORGANIZATION NAME AND ADDRESS School of Information and Computer Science Georgia Institute of Technology Atlanta, Georgia 30332	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS U. S. Army Research Office Post Office Box 12211 Research Triangle Park, NC 27709	12. REPORT DATE June 1977
	13. NUMBER OF PAGES 12
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)	15. SECURITY CLASS. (of this report) Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE NA

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

NA

18. SUPPLEMENTARY NOTES

The findings in this report are not to be construed as an official Department of the Army position, unless so designated by other authorized documents.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

graph embeddings, jumps, labels, loops, simulation, structured programming

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

It is shown that structured programs with n+1 labels can be simulated by structured programs with n labels by increasing the running time of the programs by at most a factor of 4. Two versions of this result are proved and in one version no "node splitting" operations are required in performing the simulation.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)